REMARKS

Claims 1-4, 6-11, 13-17, 19-24, and 26-27 are pending in the present application.

Claims 6 and 19 stand rejected under 35 U.S.C §112, 1st paragraph, as failing to comply with the written description requirement. Applicant respectfully traverses this rejection and respectfully requests reconsideration of the claims in light of the following remarks.

More particularly, the Examiner is unclear where, in the specification, support may be found for the limitation "different subset of bits." Applicant respectfully directs the Examiner's attention to paragraph [0019] wherein the specification discloses

> "In one implementation, the current branch address information may include some number of bits of the fetch address of the current branch instruction, for example." (Emphasis added)

In addition, at paragraph [0020] the specification discloses

> "In the illustrated embodiment, hash function blocks 230, 240 and 250 may each be configured to operate on input branch information 210 to generate a corresponding index (e.g., index 1, 2 and 3) for accessing a respective one of predictor storages 260, 270 and 280. For example, in one embodiment, hash function 230 may operate on all or part of the input branch information 210 including the global branch history and the fetch address information to generate index value 1 for accessing predictor storage 260. Likewise, hash function 240 and 250 may operate on all or part of the input branch information to generate index values 2 and 3 for accessing predictor storages 270 and 280, respectively. It is noted that each of hash function blocks 230, 240 and 250 may implement any of a variety of particular functions to generate an index value. For example, in one embodiment, hash function block 230 may perform an Exclusive-OR (XOR) function on **one portion of** the global branch information and one portion of the fetch address information. Hash function block 240 may likewise perform an XOR function on **another portion of** the global branch information and another portion of the fetch address information, and hash function 250 may perform an XOR function on **yet another portion** of the global branch information and another portion of the fetch address information." (Emphasis added)

The foregoing description teaches the branch address information includes some number of bits of the fetch address, and each hash function may operate on different portions of the fetch address. Thus it is inherent that a different portion is a different subset of bits.

Claims 1-4, 6-8, 13-17, 19-20, and 26-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Loh (U.S. Patent Publication No. 2005/0223203) (hereinafter "Loh") in view of McFarling (U.S. Patent Publication No. 2001/0056531) (hereinafter "McFarling"). Applicant respectfully traverses this rejection and respectfully requests reconsideration of the claims in light of the following remarks.

Claims 9-11 and 22-24 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Loh in view of McFarling, and in further view of Yeh (U.S. Patent No. 6,427,206) (hereinafter "Yeh"). Applicant respectfully traverses this rejection.

Applicant's claim 1 recites

> "A branch prediction mechanism comprising:
>
> a first storage including a first plurality of locations for storing a first set of partial prediction information;
>
> a second storage including a second plurality of locations for storing a second set of partial prediction information; and
>
> a control unit configured to perform a first hash function on input branch information to generate a first index for accessing a selected location within said first storage and to perform a second hash function on said input branch information to generate a second index for accessing a selected location within said second storage, wherein said input branch information includes address information corresponding to a fetch address of a current branch instruction;
>
> wherein said control unit is further configured to provide a prediction value based on corresponding partial prediction information in said selected locations of said first and said second storages; and
>
> wherein said control unit is further configured to update said selected locations of said first and said second storages dependent on whether said prediction value yields an accurate branch prediction." (Emphasis added)

The Examiner asserts the combination of Loh and McFarling teach the combination of features recited in Applicant's claim 1. Applicant respectfully disagrees with the Examiner's characterization of both Loh and McFarling, and the allegation that the combination teaches the invention recited in Applicant's claim 1.

More particularly, the Examiner alleges that Loh teaches a first storage including a first plurality of locations for storing a first set of partial prediction information (Loh Fig. 4, element 405 (e.g., branch predictor 1) and a second storage including a second plurality of locations for storing a second set of partial prediction information (Loh Fig. 4, element 405 (e.g., branch predictor 2). Applicant respectfully disagrees. Specifically, Loh teaches at paragraphs [0018], [0019], and [0022]

> "...The prediction history information <u>may be accessed in segments by a number of intermediate branch prediction units</u> 405.
>
> [0019] In one embodiment of the invention, <u>four intermediate branch history units access four segments of branch history from the branch history register.</u> However, in other embodiments, the number of segments and corresponding intermediate branch history units may be greater or fewer than four. In some embodiments of the invention, some intermediate branch history units may be in parallel and others may be in series with any of the parallel branch history units. Furthermore, the series intermediate branch history units may perform intermediate branch predictions in parallel with each other in other embodiments of the invention.
>
> [0022] FIG. 5 is a flow diagram illustrating a method for performing at least one embodiment of the invention. In operation 501, <u>a number of branch prediction segments are accessed in parallel.</u> At operation 505, <u>a number of intermediate branch predictions are performed based off of the branch prediction segments, in which each intermediate branch prediction is based off of a different branch history segment and each branch history segment is smaller than the sum of the branch history segments.</u> At operation 510, <u>a final branch prediction is made based off of the intermediate branch predictions</u>." (Emphasis added)

From the foregoing, Applicant asserts Loh merely teaches the branch predictor units 405 making predictions using information obtained from the segment registers 401, and the final predictor making a prediction based off the intermediate predictions. Applicant submiits Loh does not teach disclose or suggest that the branch predictor units

405 have any storage capacity whatsoever. Moreover Applicant cannot find any reference in Loh, whether implied or otherwise, that each predictor 405 may be a storage including a plurality of locations for storing a set of predictions. Applicant again notes that Loh does not provide any specific detail with regard to the structure of elements 405, other than what is discussed above.

Thus, Applicant submits Loh **does not teach or suggest** "a first storage including a first plurality of locations for storing a first set of partial prediction information," and "a second storage including a second plurality of locations for storing a second set of partial prediction information," as recited in Applicant's claim 1.

The Examiner acknowledges that Loh does not disclose the hashing functions as recited in Applicant's claim 1. However, the Examiner asserts McFarling teaches the limitation. Applicant respectfully disagrees. More particularly, the Examiner asserts McFarling teaches, at paragraph [0026], lines 10-13; and fig. 14, paragraphs [0087]-[0089], a control unit configured to perform a first hash function on input branch information to generate a first index for accessing a selected location within said first storage, and to perform a second hash function on said input branch information to generate a second index for accessing a selected location within said second storage.

In addition, the Examiner asserts the case of performing multiple has functions in which multiple storages can be accessed is shown. Applicant notes that while McFarling teaches branch predication, and the use of hash functions, Applicant respectfully disagrees with the Examiner's assertion that McFarling teaches the limitations as recited in Applicant's claims. It appears the Examiner is asserting that McFarling is describing the use of a single hash function (e.g., fig. 5, para [0026]), and the multiple hash functions (2) shown in fig. 14 are an extension of that. Applicant disagrees, and notes that McFarling does teach using a hash function in FIG. 5. However, Applicant asserts this is the GShare branch prediction mechanism described in Applicant's background section. Further, Applicant asserts the use of the multiple hash functions as shown in Fig. 14 of McFarling, is not a direct extension of the use of one hash function.

Specifically, McFarling teaches

> "The present invention provides a serial branch predictor that includes a first component predictor operating according to a first algorithm to predict an action, and any number of subsequent component predictors operating according to alternate algorithms to predict the action. The predictor further includes means, coupled to each predictor, for choosing between the subsequent predictors to provide a refined prediction of the action from the serial branch predictor. Such an arrangement provides a better prediction mechanism, since it serially combines multiple component predictors with varying characteristics to overrule the prediction from any prior component predictor if and only if an improvement in prediction accuracy is likely. Each subsequent stage therefore focuses on correction of predictions made by a prior stage. In the preferred embodiment, known as the SerialBLG predictor, the first predictor algorithm is bimodal, a second predictor algorithm is local, and a third predictor algorithm is global. Further, each stage is improved according to various methods." (*See* Summary) (Emphasis added)

McFarling also illustrates in FIG. 14, and teaches at paragraphs [0085]-[0091]

> "Global Stage Indexing
>
> [0086] FIG. 14 shows the global stage predictor used in the best mode SerialBLG predictor, which makes use of the stew code, conditional prediction, and partial dominance. The global information is stored in a stew register 140 as described above. The stew register value is XOR'ed with the address of the branch on line 141 to produce V on line 142: V=Stew XOR Addr.
>
> [0087] The previous stage prediction on line 143 is appended to this value to implement conditional prediction: V.sup.+=V, pred.sub.-1
>
> [0088] The value V.sup.+ provides the basis for making the global stage prediction. V.sup.30 could be used to directly access an array of counters, however, as suggested above, most of these counters would be unnecessary. A better approach is to simulate a large array of counters using a cache mechanism 144. On a cache miss, it is assumed that the unavailable count would agree with the prior stage prediction. Unavailable counters are added to the cache only if the final prediction value on line 145 is wrong.
>
> [0089] On such a replacement, the appropriate tag is set to correspond to the V.sup.30 value, and the counter 146 is initialized to weakly agree with the branch causing the miss. If the cache in the global predictor stage uses

LRU replacement, the LRU order is only affected when a counter is useful, i.e. generates a better prediction than the earlier stage.

[0090] Cache Tag Hashing

[0091] The method of indexing the global predictor stage cache is particularly significant. For good performance, set-associative caches require addresses that spread out fairly uniformly across the cache. For example, with a 2-way set-associative cache, if everything maps to the same set, then no more than two things can be stored, no matter how large the cache is. The V.sup.+ value, even using the stew code, is still less uniformly distributed than is preferred. To improve performance, two steps can be taken. First, the high order bits on line 147 of V.sup.+ are more random than the low order bits on line 148 because the high order bits on line 147 are a function of a long sequence of branches, whereas the low order bits on line 148 are a function only of the last few branches. It is therefore better to use the low order bits on line 148 for the tag value T on line 149. Second, to further increase the randomness of the remaining bits, the tag value T on line 149 can be XOR'ed into these remaining bits, Z on line 147, to obtain the set index I on line 150:..."

From the foregoing, it appears McFarling is using two hash functions (XOR) to access a single cache structure 144. In addition, Applicant notes the second hash function is operating on at least a portion of the result from the first hash function. This is in contrast to the assertions made by the Examiner, and to Applicant's invention.

Applicant further disagrees with the Examiner's assertion "It would have been obvious to use a hashing mechanism to index a set of predictors..." Applicant does not doubt the utility of using a hashing function. However, as disclosed in Applicant's background, there are index aliasing problems associated with the GShare mechanism, which Applicant's invention addresses.

Applicant submits neither Loh nor McFarling, taken either singly or in combination, teach or suggest the combination of features recited in Applicant's claim 1. Accordingly, for the reasons given above, Applicant submits claim 1, along with its dependent claims, patentably distinguishes over Loh and McFarling.

In addition, Applicant respectfully disagrees with many of the Examiner's assertions with regard to dependent claims 2-13 and 15-26. For example, in several of the dependent claim rejections, the Examiner indicates Loh discloses "the first hash function" and "the second hash function ..." Applicant submits the Examiner has already conceded that Loh does not teach any hash functions.

With regard to claim 2, the Examiner asserts Loh teaches "wherein said prediction value provides a strongly/weakly taken/not taken branch prediction indication that is indicative of whether the current branch instruction is taken upon execution." The Examiner asserts Loh teaches this at paragraph [0020]. Applicant asserts Loh ONLY teaches "a final branch history predictor unit 410 to generate <u>a final branch prediction as function of the intermediate branch predictions performed by the intermediate branch prediction units</u>." Loh is silent as to how this is performed. The Examiner is merely speculating that Loh teaches how the predictions are performed.

With regard to claims 8 and 11, the Examiner asserts that Loh discloses "using multiple values of saturating counters (typically ranging between the value of 0-3 as common at the time) and performing an action on said counters to derive a final prediction. Applicant can find absolutely no teaching of this feature in Loh. Applicant accordingly disagrees that it would not be obvious to generate "said prediction value...by summing respective counter values stored within said selected location within said first storage and said selected location within said second storage" as recited in claims 8 and 11.

Applicant's claims 14 and 27 recite features that are similar to the features recited in Applicant's claim 1. Thus, for at least the reasons given above, Applicant submits claims 14 and 27, along with their respective dependent claims, patentably distinguish over Loh and McFarling.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-96100/BNK.

Respectfully submitted,

Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: January 30, 2007